

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
1/13

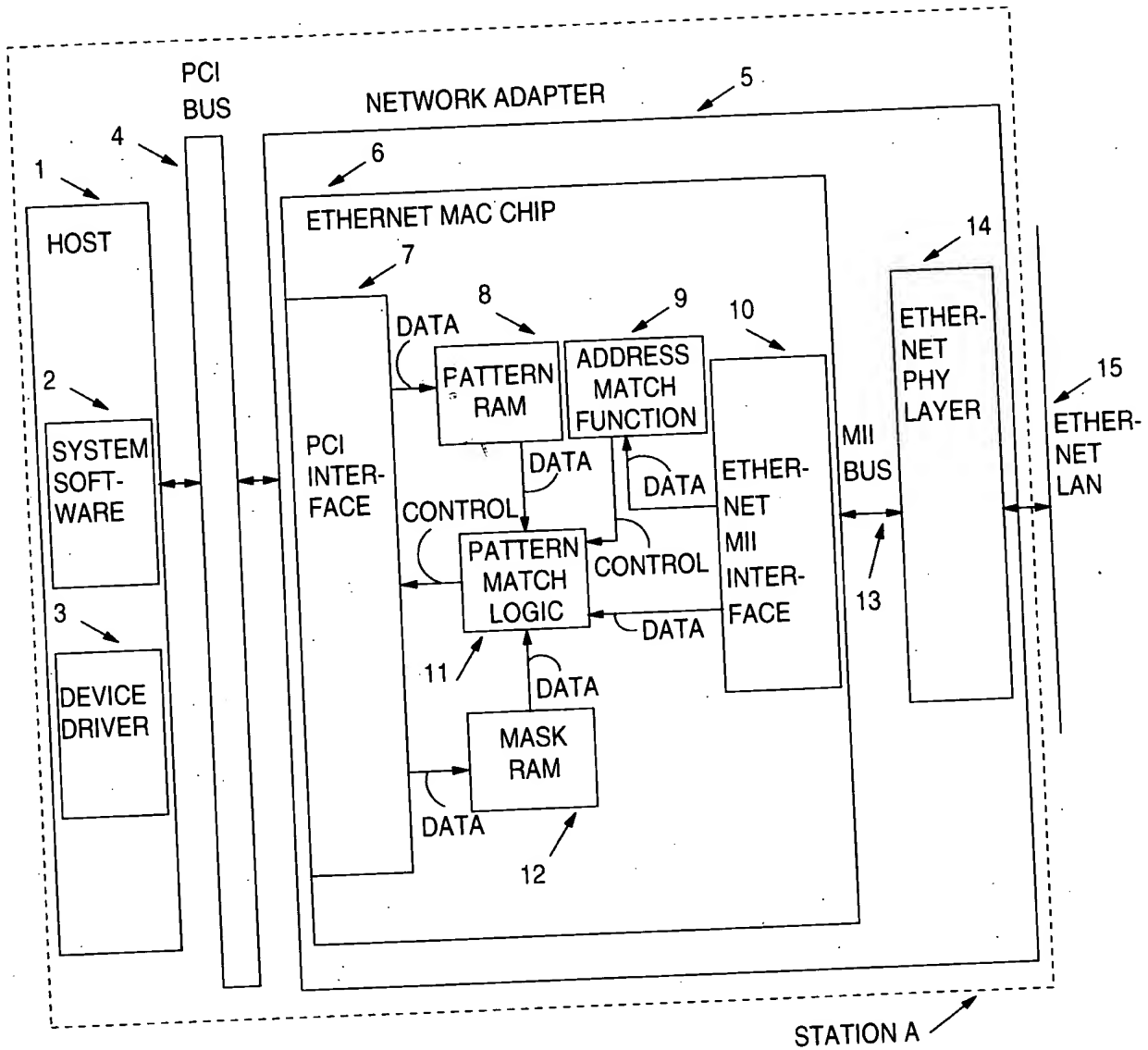


FIG. 1

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
2/13

DA	SA	LENGTH/ TYPE	DATA	PADDING (IF NEEDED)	CRC
----	----	-----------------	------	------------------------	-----

DA - DESTINATION ADDRESS (6 BYTES)

SA - SOURCE ADDRESS (6 BYTES)

LENGTH/TYPER - LENGTH OF DATA FIELD (IEEE 802.3)/TYPE DEFINITION (ETHERNET)
(2 BYTES)

DATA - LLC DATA (INCLUDING PADDING, IF NEEDED, IN SHORT LLC FRAMES)
(46 TO 1500 BYTES)

CRC - CYCLE REDUNDANCY CHECK (4 BYTES)

ETHERNET FRAME FORMAT

FIG. 2

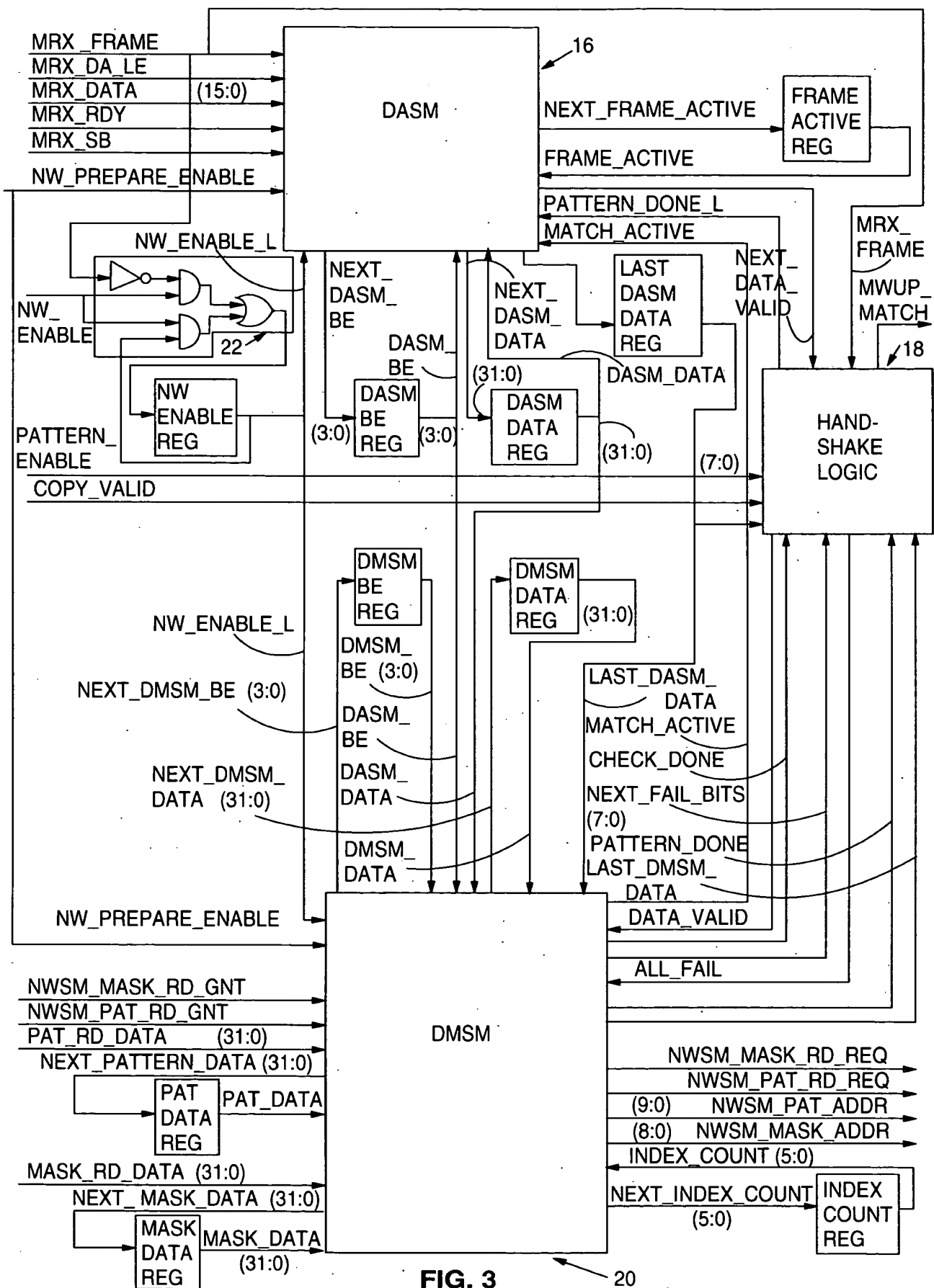


FIG. 3

20

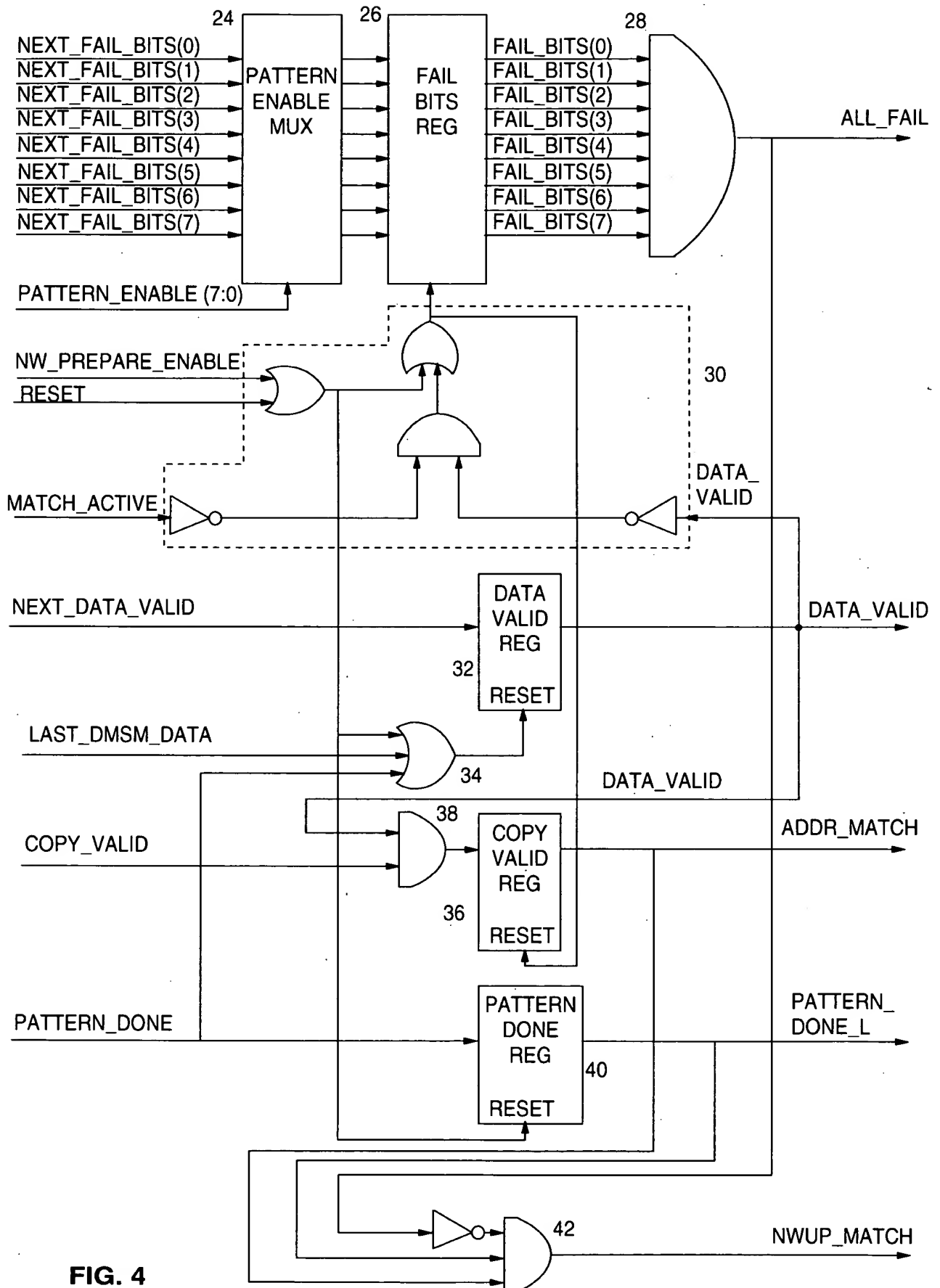



FIG. 4


APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
5/13



DMSM_STATE(2:0) INDEX_COUNT

FIG. 5A



DMSM_STATE(2:0) INDEX_COUNT

FIG. 5B

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
6/13

STATE 0: IDLE/DATA31:16

IF PREPARE FOR ENABLE IS SET, OR ENABLE IS TURNED OFF

TURN OFF DATA_VALID AND FRAME_ACTIVE AND STAY IN STATE 0

IF FRAME FROM MEDIA ENDS (MRX_FRAME IS DEASSERTED) AND FRAME_ACTIVE FLAG IS SET

TURN OFF FRAME_ACTIVE FLAG, MARK DATA AS LAST DATA TRANSFER TO DMSM (TURN ON LAST_DASM_DATA), AND STAY IN STATE 0

IF FRAME IS BEING RECEIVED (MRX_FRAME IS ASSERTED), AND FRAME_ACTIVE IS NOT SET (THIS MEANS THE FRAME IS JUST STARTING), AND DESTINATION ADDRESS LATCH ENABLE IS SET (MRX_DA_LE IS ASSERTED), AND MRX_RDY IS SET (I.E. 16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MATCH_ACTIVE = 0 (DMSM IS NOT CURRENTLY FINISHING MATCHING ON A PREVIOUS FRAME), AND MRX_SB IS NOT ASSERTED (I.E. IT'S A FULL 2 BYTE TRANSFER)

SAVE DATA FROM MRX_DATA IN DASM_DATA(31:16), SET FRAME_ACTIVE SIGNAL TO SHOW WE'RE RECEIVING A FRAME. SET DASM_BE (VALID BYTES BIT MASK) TO '1100' AND GO TO STATE 1 TO GET BITS 15:0

IF FRAME IS BEING RECEIVED (MRX_FRAME IS ASSERTED), AND FRAME_ACTIVE IS NOT SET (THIS MEANS THE FRAME IS JUST STARTING), AND DESTINATION ADDRESS LATCH ENABLE IS SET (MRX_DA_LE IS ASSERTED), AND MRX_RDY IS SET (I.E. 16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MATCH_ACTIVE = 0 (DMSM IS NOT CURRENTLY FINISHING MATCHING ON A PREVIOUS FRAME), AND MRX_SB IS ASSERTED (I.E. IT'S A 1 BYTE TRANSFER, WHICH CAN ONLY HAPPEN AT THE END OF A FRAME)

STATE 1: DATA15:0

IF PREPARE FOR ENABLE IS SET, OR ENABLE IS TURNED OFF

TURN OFF DATA_VALID AND FRAME_ACTIVE AND GO TO STATE 0

IF FRAME FROM MEDIA ENDS (MRX_FRAME IS DEASSERTED) AND FRAME_ACTIVE FLAG IS SET

TURN OFF FRAME_ACTIVE FLAG, MARK DATA AS LAST DATA TRANSFER TO DMSM (TURN ON LAST_DASM_DATA), SET DATA_VALID TO TELL DMSM TO PROCESS THE DATA IN DASM_DATA, AND GO TO STATE 0

IF PATTERN_DONE IS SET (I.E. DMSM HAS FINISHED CHECKING ALL PATTERNS)

GO TO STATE 0

IF FRAME IS BEING RECEIVED AND FRAME_ACTIVE IS SET AND MRX_RDY IS ASSERTED (16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MRX_SB IS NOT ASSERTED (I.E. IT'S A FULL 2 BYTE TRANSFER)

FIG. 6A

FIG. 6A
FIG. 6B

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
7/13

STATE 0: IDLE/DATA31:16

SAVE DATA FROM MRX_DATA IN DASM_DATA(31:16), SET FRAME_ACTIVE SIGNAL TO SHOW WE'RE RECEIVING A FRAME. SET DASM_BE (VALID BYTES BIT MASK) TO "1000", SET DATA_VALID TO TELL DMSM TO PROCESS THE DATA IN DASM_DATA, AND STAY IN STATE 0, AS THIS WAS LAST DATA TRANSFER FOR THE FRAME

IF PATTERN_DONE IS SET (I.E. DMSM HAS FINISHED CHECKING ALL PATTERNS)

STAY IN STATE 0

IF FRAME IS BEING RECEIVED AND FRAME_ACTIVE IS SET AND MRX_RDY IS ASSERTED (16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MRX_SB IS NOT ASSERTED (I.E. IT'S A FULL 2 BYTE TRANSFER)

SAVE DATA FROM MRX_DATA IN DASM_DATA(31:16), SET DASM_BE TO "1100", AND GO TO STATE 1

IF FRAME IS BEING RECEIVED AND FRAME_ACTIVE IS SET AND MRX_RDY IS ASSERTED (16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MRX_SB IS ASSERTED (I.E. IT'S A 1 BYTE TRANSFER, AND LAST TRANSFER OF FRAME)

SAVE DATA FROM MRX_DATA IN DASM_DATA(31:16), SET DASM_BE TO "1000", SET DATA_VALID TO TELL DMSM TO PROCESS THE DATA IN DASM_DATA, AND STAY IN STATE 0

ELSE

STAY IN STATE 0

STATE 1: DATA15:0

SAVE DATA FROM MRX_DATA IN DASM_DATA(15:0), SET DASM_BE TO "1111", SET DATA_VALID TO TELL DMSM TO PROCESS THE DATA IN DASM_DATA, AND GO TO STATE 0

IF FRAME IS BEING RECEIVED AND FRAME_ACTIVE IS SET AND MRX_RDY IS ASSERTED (16 BITS OF DATA IS BEING TRANSFERRED FROM THE MEDIA) AND MRX_SB IS ASSERTED (I.E. IT'S A 1 BYTE TRANSFER, AND LAST TRANSFER OF FRAME)

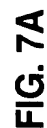
SAVE DATA FROM MRX_DATA IN DASM_DATA(15:0), SET DASM_BE TO "1110", SET DATA_VALID TO TELL DMSM TO PROCESS THE DATA IN DASM_DATA, AND GO TO STATE 0

ELSE

STAY IN STATE 1

FIG. 6B

FIG. 6A
FIG. 6B



STATE 0: IDLE

STATES 1, 2, 3, 4, 5, 6, 7: PATTERNS 1-7

STATE 8: PATTERN 8

SET THE FAIL BIT FOR THE PATTERN NUMBER BEING CHECKED (STATE NUMBER - 1) READ THE PATTERN DATA FOR THE NEXT PATTERN (ADDRESS IS STATE (2:0) AND INDEX COUNT) AND ASSERT MATCH_ACTIVE FLAG TO TELL DASM THAT MATCHING IS ACTIVE, AND TO GO TO NEXT SEQUENTIAL STATE (I.E. IF IN STATE 1, GO TO STATE 2...IF IN STATE 7, GO TO STATE 8). NOTE: IF IN STATE 7 INCREMENT INDEX COUNT BEFORE GOING TO STATE 8, ELSE, LEAVE IT AS IS.

SET FAIL BIT FOR PATTERN 8 (FAIL BIT 7) READ PATTERN DATA FOR PATTERN 1 (ADDRESS IS STATE (2:0) & UPDATED INDEX COUNT), READ NEXT MASK WORD (ADDRESS IS UPDATED INDEX COUNT), TRANSFER DATA FROM DASM_DATA REGISTER TO DMSM_DATA REGISTER. TRANSFER DATA VALID BITS FROM DASM_BE TO DMSM_BE. GO TO STATE 1 TO START CHECK OF ALL 8 PATTERNS ON THIS NEW WORD OF DATA FROM DASM. IF THIS IS LAST TRANSFER OF A FRAME (FRAME HAS ENDED BEFORE ALL 128 BYTES HAVE BEEN CHECKED), SET LAST TRANSFER FLAG.)

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL IS NOT ASSERTED), THEN IF DATA_VALID IS SET (DMSM HAS VALID DATA FROM DASM), AND THE CURRENT PATTERN'S DATA WORD MATCHES THE DATA FROM THE MEDIA (I.E. THE DATA IN DMSM) FOR BITS THAT ARE SET IN THE MASK (I.E. THIS PATTERN HAS FAILED MATCHING)

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL IS NOT ASSERTED), THEN IF DATA_VALID IS SET (DMSM HAS VALID DATA FROM DASM), AND PATTERN 8'S DATA WORD DOES NOT MATCH THE DATA FROM THE MEDIA (I.E. THE DATA IN DMSM_DATA) FOR BITS THAT ARE SET IN THE MASK BITS 31:28 (I.E. PATTERN 8 HAS FAILED MATCHING) AND INDEX COUNT IS EQUAL TO 32

READ THE PATTERN DATA FOR THE NEXT PATTERN (ADDRESS IS STATE & INDEX COUNT) AND ASSERT MATCH_ACTIVE FLAG TO TELL DASM THAT MATCHING IS ACTIVE, AND GO TO NEXT SEQUENTIAL STATE (I.E. IF IN STATE 1, GO TO STATE 2...IF IN STATE 7, GO TO STATE 8) NOTE: IF IN STATE 7, INCREMENT INDEX COUNT BEFORE GOING TO STATE 8, ELSE, LEAVE IT AS IS.

SET FAIL BIT FOR PATTERN 8 (FAIL BIT 7), SET PATTERN_DONE FLAG TO TELL DASM THAT ALL PATTERN MATCHING IS COMPLETE, AND GO TO STATE 0 TO AWAIT DATA FROM NEXT FRAME.

FIG. 7A
FIG. 7B
FIG. 7C
FIG. 7D

FIG. 7B

STATE 0: IDLE

STATES 1, 2, 3, 4, 5, 6, 7: PATTERNS 1-7

STATE 8: PATTERN 8

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL IS NOT ASSERTED), THEN IF DATA_VALID IS NOT SET (FRAME HAS ENDED BEFORE DMSM HAS CHECKED ALL 128 BYTES OF ALL PATTERNS), DMSM MUST NOW MAKE SURE ALL MASK BITS ARE TURNED OFF FOR THIS PATTERN - IF ANY MASK BITS FOR THIS PATTERN ARE ON...

SET THE FAIL BIT FOR THE PATTERN NUMBER BEING CHECKED (STATE NUMBER - 1), AND GO TO NEXT SEQUENTIAL STATE. IT IS NO LONGER NECESSARY TO READ PATTERN DATA FROM THE PATTERN RAM. NOTE: IF IN STATE 7 INCREMENT INDEX COUNT BEFORE GOING TO STATE 8, ELSE LEAVE IT AS IS.

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL IS NOT ASSERTED) THEN IF DATA_VALID IS SET (DMSM HAS VALID DATA FROM DASM), AND PATTERN 8'S DATA WORD DOES MATCH THE DATA FROM THE MEDIA (I.E. THE DATA IN DMSM_DATA) FOR BITS THAT ARE SET IN THE MASK BITS 31:28 (I.E. PATTERN 8 HAS FAILED MATCHING) AND INDEX COUNT IS EQUAL TO 32

SET PATTERN_DONE FLAG TO TELL DASM THAT ALL PATTERN MATCHING IS COMPLETE, AND GO TO STATE 0 TO AWAIT DATA FROM NEXT FRAME.

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL IS NOT ASSERTED) THEN IF DATA_VALID IS SET (DMSM HAS VALID DATA FROM DASM), AND PATTERN 8'S DATA WORD DOES NOT MATCH THE DATA FROM THE MEDIA (I.E. THE DATA IN DMSM_DATA) FOR BITS THAT ARE SET IN THE MASK BITS 31:28 (I.E. PATTERN 8 HAS FAILED MATCHING) AND INDEX COUNT IS LESS THAN 32 (WE'VE NOT CHECKED ALL 128 BYTES OF THE PATTERN YET)

FIG. 7A
FIG. 7B
FIG. 7C
FIG. 7D

FIG. 7C

STATE 0: IDLE

STATES 1, 2, 3, 4, 5, 6, 7: PATTERNS 1-7

STATE 8: PATTERN 8

READ PATTERN DATA FOR PATTERN 1
(ADDRESS IS STATE(2:0) & UPDATED INDEX
COUNT) READ NEXT MASK WORD (ADDRESS IS
UPDATED INDEX COUNT), TRANSFER DATA
FROM DASM_DATA REGISTER TO DMSM_DATA
REGISTER. TRANSFER DATA VALID BITS FROM
DASM_BE TO DMSM_BE. GO TO STATE 1 TO
START CHECK OF ALL 8 PATTERNS ON THIS
NEW WORD OF DATA FROM DASM. IF THIS IS
LAST TRANSFER OF A FRAME (FRAME HAS
ENDED BEFORE ALL 128 BYTES HAVE BEEN
CHECKED), SET LAST TRANSFER FLAG).

IF ALL PATTERNS HAVE NOT FAILED (ALL_FAIL
IS NOT ASSERTED), THEN IF DATA_VALID IS
NOT SET (FRAME HAS ENDED BEFORE DMSM
HAS CHECKED ALL 128 BYTES OF ALL
PATTERNS), DMSM MUST NOW MAKE SURE ALL
MASK BITS FOR PATTERN 8 (BITS 31:28) ARE
TURNED OFF. IF ANY OF THESE BITS ARE ON...

SET THE FAIL BIT FOR THE PATTERN 8 IF INDEX
COUNT IS LESS THAN 32, READ NEXT WORD OF
MASK (ADDRESS IS UPDATED INDEX COUNT)
AND GO TO STATE 1

IF INDEX COUNT IS EQUAL TO 32, SET
PATTERN_DONE AND GO TO STATE 0

FIG. 7A
FIG. 7B
FIG. 7C
FIG. 7D

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

RA9-98-007
ALLISON ET AL
12/13

PATTERN NUMBER	PATTERN RAM LOCATION (BYTE ADDRESS)	PATTERN RAM LOCATION (WORD ADDRESS)
1	000-07F	000-01F
2	080-0FF	020-03F
3	100-17F	040-05F
4	180-1FF	060-07F
5	200-27F	080-09F
6	280-2FF	0A0-0BF
7	300-37F	0C0-0DF
8	380-3FF	0E0-0FF

FIG. 8

APPROVED	U.S. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

ALLISON ET AL
13/13

BITS

31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0
MASK FOR PATTERN 8 WORD 1	MASK FOR PATTERN 7 WORD 1	MASK FOR PATTERN 6 WORD 1	MASK FOR PATTERN 5 WORD 1	MASK FOR PATTERN 4 WORD 1	MASK FOR PATTERN 3 WORD 1	MASK FOR PATTERN 2 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1	MASK FOR PATTERN 1 WORD 1

FIG. 9